한국컴퓨터정보학회 하계학술대회 논문집 제28권 제2호 (2020. 7)

발행-구독 프로토콜에서 전송 성능의 비교 및 분석

판주지에[O], 김재수[*]
[O]경북대학교 컴퓨터학과,
[*]경북대학교 컴퓨터학과
e-mail: 2020224076@knu.ac.kr[O], kjs@knu.ac.kr[*]

Transmission Performance Comparison and Analysis with Different Publish/Subscribe Protocol

Zujie Fan[O], JaeSoo Kim[*]
[O]School of Computer Science and Engineering, Kyungpook National University,
[*]School of Computer Science and Engineering, Kyungpook National University

● 요 약 ●

In this paper, we analyze and compare the performance of different publish and subscribe protocols in the real application environment. This paper provides a horizontal comparison of current publish/subscribe protocols in terms of security, throughput, and delay performance. Thanks to the use of lightweight frameworks, the MQTT protocol has demonstrated excellent performance in terms of delay performance. However, the AMQP protocol has more advantages in security and throughput. Although the REST/HTTP protocol has the worst delay performance, it is excellent in terms of compatibility because it is based on the HTTP protocol.

키워드: Internet of Things, MQTT, AMQP, REST/HTTP, CoAP, Publish/Subscribe Protocol

I. Introduction

Today IoT (Internet of Things) is used in various fields, but the IoT architecture is still built on the traditional Internet infrastructure. In the current internet communication, due to the low development cost of the HTTP (HyperText Transfer Protocol) protocol, the degree of openness is high. It's very attractive to many manufacturers. However, for frequent operation scenarios or some terminals with poor performance, the implementation cost and real-time performance of the HTTP protocol are unsatisfactory. With the development of IoT applications, there are a lot of publishing and subscription protocols for developers to choose from. For example, AMQP (Advanced Message Queuing Protocol), JMS (Java Message Service), REST(Representational State Transfer)/HTTP work in Ethernet, CoAP (Constrained Application Protocol) protocol is a protocol especially developed for resource-constrained devices, and DDS (Data Distribution Service) for Real-Time Systems) and MQTT (Message Queuing Telemetry Transport) are applicable to all IoT environments, each communication protocol has its own advantages and weakness. According to the communication needs of the actual scenario, the appropriate protocol should be chooses. How to choose among many protocols has become an open issue for development based on IoT devices[1]. Such as MQTT protocol, CoAP protocol can be used for home automation systems[2], CoAP, and MQTT can be used for hand-held medical information collection equipment, and implementation of REST/HTTP in dynamic wireless sensor[6]. DDS protocol is suitable for data communication in high reliability and high security applications, because it performs poorly on wireless networks, especially when resources are limited, DDS protocol is widely used in industrial control systems[7]. Therefore, it is difficult to choose the best protocol for all devices. In order to solve these issues, this paper selected several popular protocols (MQTT, CoAP, AMQP, REST/HTTP) to make a performance comparison and analysis under Raspberry Pi and PC platforms.

II. Related Works

Table1 compares the different publish/subscribe protocol under the characteristics of abstract, transport layer, QoS and security.

2.1 MQTT

MQTT is an instant messaging protocol designed by IBM. It is lightweight, simple, open, easy to implement, and supports all platforms [3]. It is suitable for low bandwidth and unreliable network.

2.2 CoAP

CoAP is a RESTful API that simplifies the HTTP protocol, it's a protocol supports asynchronous communication, which is a common sleep/wake mechanism for M2M(Machine to Machine) communication applications[4], it makes up for the disadvantage that HTTP is not suitable for M2M communication. So it is suitable for IP networks with limited resource communication.

2.3 AMQP

AMQP is mostly used for data exchange in business systems such as PLM (Product Lifecycle Management), ERP (Material Requirement Planning), MES (Manufacturing Execution System). In IoT applications, it is mainly suitable for communication and analysis of mobile handheld devices and back-end data centers.

2.4 REST/HTTP

REST/HTTP is a communication style developed based on the HTTP protocol, it's mainly to simplify the system architecture in the Internet, quickly realize the loose coupling between the client and the server, and reduce the interaction delay between the client and the server. It is suitable for the application level of the Internet of Things, and the resources in the Internet of Things are opened through REST, so that the service is called by other applications.

III. Protocol Evaluation Model

The Publish/subscribe model is a one-to-many communication mode in which multiple subscriber objects listen simultaneously to a topic object that notifies all subscribers when its state changes, enabling them to automatically update their state. Fig 1 shows a basic publish/subscribe architecture.

Table 1. The comparison of Publish/Subscribe Protocols

Protocol	Abstract	Transport layer	Qos	Security
MQTT	Pub/Sub	TCP	Level match	Username and Password SSL/TLS
CoAP	Request /Reply	UDP	Multicast address	NoSec mode
AMQP	Pub/Sub	TCP	Queue and message filtering	SSL Certification TLS Data Encryptio
REST/ HTTP	Request/ Reply	TCP	N/a	SSL/TLS

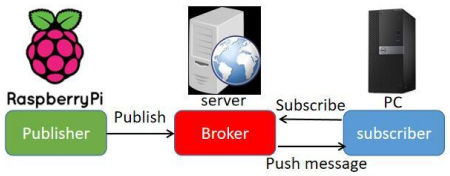

Fig. 1. Publish / Subscribe Architecture

Table 2. Experimental Environment

Protocol	Publisher	Broker	Subscriber
MQTT	Raspberry Pi (python)	PC(mosquitto) [2]	PC(python)
CoAP	Raspberry Pi (python)	Raspberry Pi (CoAPthon)	EMQ X Mqtt.fx[5]
AMQP	Raspberry Pi (python)	Raspberry Pi (Rabbitmq)[8]	PC (python)
REST/ HTTP	PC1. restclient	PC1. jersey	PC2. restclient

This paper uses the publish/subscribe model for data communication testing. Raspberry Pi and PC are used as test platforms. Table 2 shows the experimental environment. This section will introduce the experimental process and problem analysis of each protocol.

In this experiment, the delay D of real-time communication of each protocol is counted, through protocol communication, the time of sending a message is directly sent to the receiver as a message, the receiving end receives the data and records the current receiving time, through the formula (1), let k =10, subtract to obtain the delay time.

$$D = \frac{1}{k}\sum_{i=1}^{k}(T_{receive} - T_{send}) \qquad (1)$$

MQTT: The MQTT protocol can be implemented in multiple languages. To ensure the fairness of the experiment results, all communication testing are based on python language and using mosquitto as the server, PC as the the receiver, and raspberry pi as the subscriber for experiment data collection.

CoAP: The CoAPthon library is adopted as a server and raspberry pi as a sender, EMQ X (Erlang/Enterprise/Elastic MQTT Broker) message server extension support CoAP protocol, therefore, in order to ensure fast and stable service, using the receiver of the message server as a message receiver.

AMQP: It is based on RabbitMQ in Erlang language, OpenAMQ in C language, the Apache Qpid is developed by Apache and StormMQ message queue based on netty language. The RabbitMQ are adopted as the server, because its performance was relatively better and more comprehensive [8].

REST/HTTP: It is a service call API encapsulation style in the IoT system, through the way of opening REST APIs, open data service to achieve data exchange. In this experiment, a server was built through jersey framework, and restclient was used as the receiving end to conduct experimental measurement. All experimental data were averaged by multiple experiments.

IV. Performance comparison and analysis

Fig 2 shows their delay performance comparison. We can draw a conclusion from the experimental results, the delay of the MQTT protocol is the lowest, the AMQP protocol is slightly slower than the MQTT protocol, the CoAP protocol has higher latency, and the REST/HTTP protocol has the highest latency.

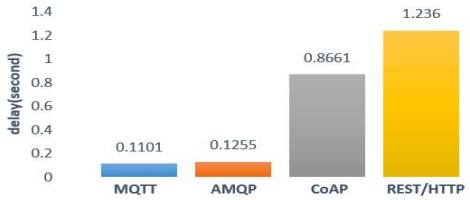

Fig. 2. Delay Performance Comparison

Fig 3 shows their throughput performance comparison. The transmission throughput F is calculated by formula (2), where VU is the number of clients, R is the number of requests issued by clients, and T is the time spent on the performance test. We can draw a conclusion from the experimental results, the throughput performance of the AMQP protocol is the worst, the REST/HTTP protocol is slightly better than the AMQP protocol, the CoAP protocol has better throughput performance, and the MQTT protocol has the best performance of throughput.

$$F_{avg} = \frac{1}{k}\sum_{i=1}^{k}\frac{VU \times R}{T} \qquad (2)$$

Fig. 3. Throughput Performance Comparison

Table 3 shows the safety level comparison. The MQTT utilized TLS (Transport Layer Security) encryption on the transport layer to ensure security, hence, MQTT security mainly depends on the broker, CoAP is secured with DTLS(Datagram Transport Layer Security) encryption, which is only applicable for unicast. AMQP adopted RabbitMQ as a broker and SSL certificates are used by default. The REST/HTTP protocol is just one communication style of HTTP and security depends on the HTTP protocol, with the development of IoT technology, the security of these protocols is getting more and more attention[9].

Table 3. Safety level comparison

Protocol	Excellent	Good	Average	Poor
MQTT				●
CoAP			●	
AMQP	●			
REST/ HTTP		●		

V. Conclusions

This paper conducted performance comparison experiments on four publish/subscribe protocols, and considered the security and reliability. meanwhile, this paper compare and analyze the suitability of these protocols in real application environment. We draw the following conclusions, MQTT protocol is more suitable for low latency, and relatively low safety factor working environment, because MQTT is lightweight, suitable for real-time monitoring application environment. it can significantly reduce energy consumption and cost of development. The AMQP protocol has an excellent performance in terms of security, it's suitable for data exchange of gateways of IoT devices exposed

to the Internet. Although the performance of REST / HTTP is the worst in delay, in the IoT data acquisition into the IoT application system performance is relatively outstanding, through open REST APIs, it can make other applications obtain better interactivity.

REFERENCES

[1] Sufiya S Kazi, Gayatri Bajantri, and Trupti Thite, "Remote Heart Rate Monitoring System Using IoT," International Research Journal of Engineering and Technology(IRJET) Vol. 05, No. 4, pp. 1-8, April 2018.

[2] Yuvraj Upadhya, Mr.Amol Borole, and Mr.D.Dileepan, "MQTT Based Secured Home Automation System," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), pp. 1-4, March 2016.

[3] Dipa Soni, Ashwin Makwana, ”A Survey On Mqtt: A Protocol Of Internet Of Things(Iot),”International Conference on Telecommunication, Power Analysis and Computing Techniques)ICTPACT - 2017) pp.2, April 2017.

[4] Z. Shelby, K. Hartke, C. Bormann, ”The Constrained Application Protocol (CoAP)” Internet Engineering Task Force (IETF), pp.10, 2014.

[5] EMQ X, https://www.emqx.io/downloads#enterprise.

[6] Cao Vien Phung, Jasenka Dizdarevic, Admela Jukan, ”An Experimental Study of Network Coded REST HTTP in Dynamic IoT Systems” IEEE International Conference on Communications (ICC), Dublin, Ireland pp.1-6, Fe 2020.

[7] Tarek A. Youssef, Mohamad El Hariri, Ahmed T. Elsayed, Osama A. Mohammed, ”A DDS-Based Energy Management Framework for Small Microgrid Operation and Control” IEEE Transactions on Industrial Informatics, Vol. 14, pp.1-11, September 2017.

[8] RabbiMQ, https://www.rabbitmq.com

[9] S. Zamfir, T. Balan, I. Iliescu, F. Sandu, ”A Security Analysis on Standard IoT Protocols” International Conference on Applied and Theoretical Electricity (ICATE), pp.1-6, November 2016.